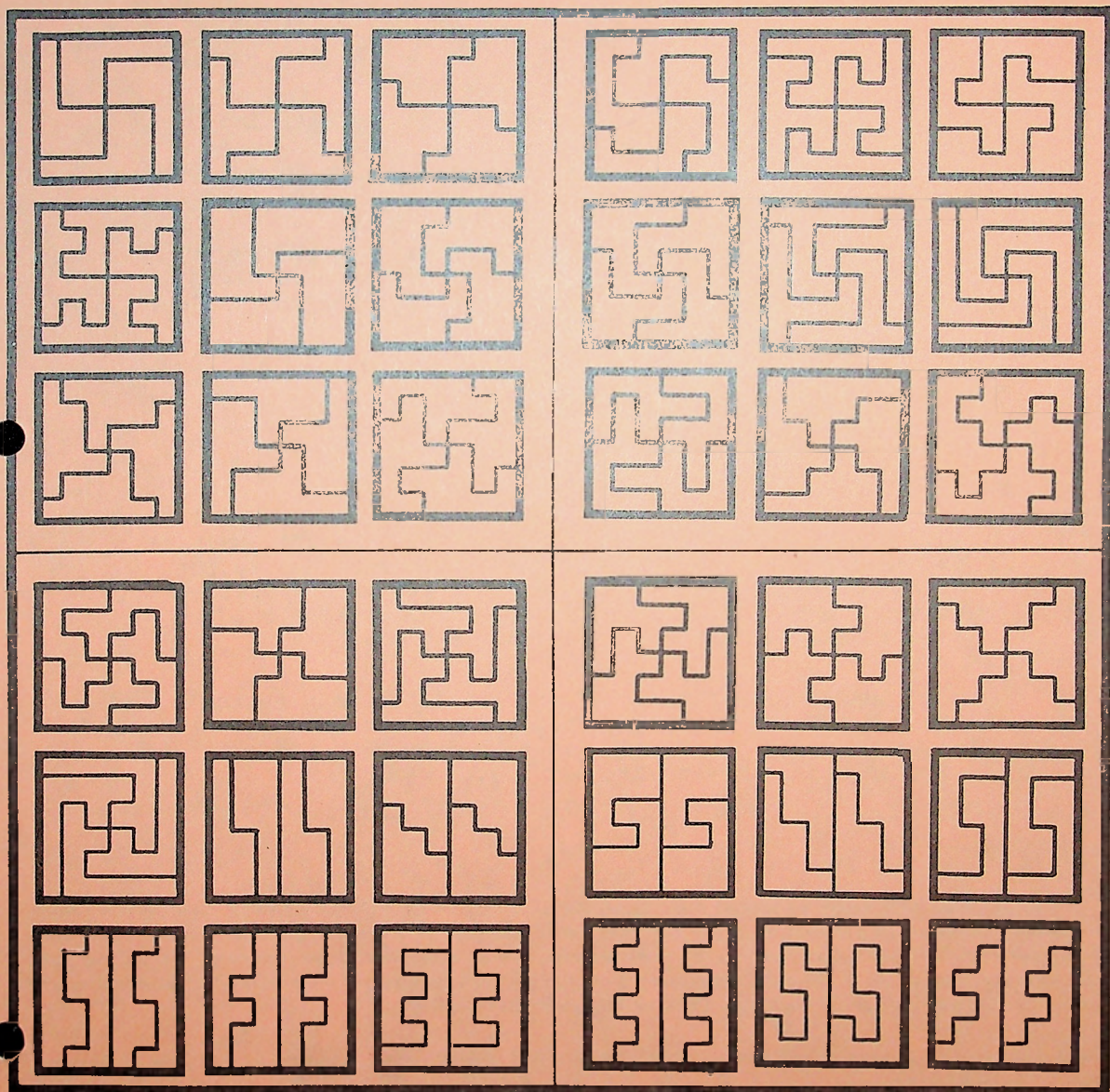# Checkerboard

PROBLEM 15

The Checkerboard problem is another of those that first appeared in the Graham Dial magazine some 30 years ago; it is No. 44 in the book Ingenious Mathematical Problems and Methods (L. A. Graham, Dover Publications, 1959). The problem, as given then, was to find the number of ways in which a 6 x 6 checkerboard could be cut into four congruent pieces, following the lines of the board. The Figure on the cover of this issue shows the 37 solutions.

The solution that appeared in the Dial magazine showed 95 ways of making the cuts, but these included simple rotations and all mirror images. The 37 solutions given here are all unique.

For a 2 x 2 checkerboard, there is only one way to cut it into four congruent pieces. For a 4 x 4 board, there are just 5 unique ways. Thus, we have the table of known values:

N	n
2	1
4	5
6	37

What can we expect for the 8 x 8 board? Casual analysis, by hand, suggests that there are around 250 ways to cut up the board. Could this number be determined precisely with a computer program? If so, the program should be generalized to handle the 10 x 10 case and higher.

Cubic curve fitting of the available data (using 250 for the 8 x 8 case) indicates that the 10 x 10 result, when it is found, may be around 800, and the 12 x 12 result may be around 1800.

Pattern Recognition

Articles in the popular press tend to indicate that pattern recognition by computer program, if not already an accomplished fact, is about ready to emerge. Programs are almost perfected, it would seem, for things like fingerprint matching, voice recognition, and signature identification. Actually, these goals are elusive, and may be far off. Try a simple problem in pattern recognition.

A block of storage, B, contains 1000 decimal digits. (The block could be 1000 consecutive words, with each word containing one digit in 4-bit BCD form.) We wish to search the 1000 digits and locate:

(1) The longest run of odd digits.

(2) The longest run of even digits.

(3) The longest run of digits of the form EEOOEEOO...

(4) The longest run of a repeated digit; for example, ...66666666...

(5) The longest run of digits in the natural order 01234567890123456...

(A) Draw a flowchart for this logic.

(B) Write a program in any language to input the 1000 digits, 50 to a card, and output the location in the block of the five required sets.

For example, for the 1000 digits of pi (PC6, page 3) we have the following results (the starting position is given in each case):

(1) 11 odd digits start at 940.
(2) 10 even digits start at 70.
(3) 8 digits of the form EEOOEE...start at 89.
(4) 6 of the digit 9 repeat at 763.
(5) 4 digits in natural order start at 659.

The June, 1973 issue of <u>Consumer Reports</u> contained a review of 15 pocket calculators. The article may be purchased from Consumers Union Back Issue Department, Orangeburg, NY 10962, for 60¢. The article and review is done with their usual thoroughness, including measures of battery life, recharge time, and the viewing angle of the display for each machine. The machines reviewed all sell for under $130, and are all simple calculators; that is, none has functions beyond the four arithmetic processes.

Speaking of Languages

BY ROBERT TEAGUE

It seems appropriate to turn our attention this month to a programming problem.   Although other languages will be featured in this column, the first programming problem is in Fortran, since this is still the most common language in use in the United States.

The Change Maker is a very old problem.   Basically, it consists of inputting a value to represent the amount purchased by a fictional customer, and also the amount tendered in payment.   From this the proper change, in terms of the least number of coins and bills, is computed and printed.   Two examples follow.

```
AMOUNT PURCHASED $  92
AMOUNT TENDERED  $ 200

BROKEN DOWN AS FOLLOWS-

      1 DOLLAR BILL(S)
      0 HALF
      0 QUARTER
      0 DIME(S)
      1 NICKEL
      3 PENNY(IES)


AMOUNT PURCHASED $ 109
AMOUNT TENDERED  $ 200

BROKEN DOWN AS FOLLOWS-

      0 DOLLAR BILL(S)
      1 HALF
      1 QUARTER
      1 DIME(S)
      1 NICKEL
      1 PENNY(IES)
```

This exact output was produced by a Fortran program of only 11 statements.   It is a proper ANSI code, including the END, etc.   The program is written with a loop to process more than one input, and appears to be solved with the least number of statements for a program using a loop.   The shortest program to solve the problem for a single input value appears to be 9 statements.

Admittedly, the program with the fewest statements is not necessarily the most efficient program, for there are other variables involved in efficiency, including storage requirements, execution time, etc.   But tight code is one measure, and helps to indicate the level of sophistication of the programmer.   It is also fun at times to match wits and programming skill with others of equal competence.

A challenge, then---can you solve this problem so that it produces the output shown above in fewer than 11 Fortran statements?   Only proper ANSI standard Fortran codes, making use of no special features of any machine, are to be considered here.   Checkout will be done on a CDC 3300.   The person with the fewest statements will be declared the winner, and his solution will be published in this column.   Send your output to:

Speaking of languages...
POPULAR COMPUTING
Box 272
Calabasas, CA   91302

The following Fortran program furnishes the day of the week for a given date in the range covered by our current calendar.   For the results, 0 is Sunday, 1 is Monday, 2 is Tuesday,..., 6 is Saturday.

```
10  10      READ(0,0) MO,K,IYR
20          M= MOD(MO+9,12)+1
30          ID = IYR-M/11
40          IC = ID/100
50          ID = MOD (ID,100)
60          J = MOD(IFIX(2.6*FLOAT(M)-0.2)+K+ID+ID/4+IC/4-2*IC,7)
70          WRITE(1,0) MO,K,IYR,J
80          GO TO 10
90          END
READY
RUN
12,7,1941
12,7,1941,0
11,11,1991
11,11,1991,1
10,15,1973
10,15,1973,1
11,15,1973
11,15,1973,4
12,25,1973
12,25,1973,2
1,1,1800
1,1,1800,3
1,1,1900
1,1,1900,1
```

Feedback

The Fourway problem (problem No. 5) first appeared in issue 3.   A square array of cells each contains a number, from 1 to 4.   The number indicates a direction to follow; namely, 1 means up; 2, right; 3, down; and 4, left.   After moving in the direction dictated by the number in the cell, the number is incremented by one, modulo 4.   A single game is played by starting at the center cell of the array and moving until an exit from the array takes place.   For the 3 x 3 case, starting with 1's in all nine cells, the original pattern returns after 16 games.   This is what is now known about Fourway:

Case	Cycle length
3 x 3	16
5 x 5	104
7 x 7	544
9 x 9	146248
11 x 11	(greater than 1030769)

The last partial result is furnished by Tom Cundey.   Thus, the cycle length for the 11 x 11 case is still not known, but it is evidently enormous.   Of course, the fact that the cases lower than 11 x 11 do cycle back to the original pattern is no proof that higher cases will.

Meanwhile, Mr. Cundey has analyzed some three dimensional cases, as follows:

3 x 3 x 3	6-way	102	Proceeding through the faces of the cubes
5 x 5 x 5	6-way	2376	
3 x 3 x 3	8-way	56	Proceeding through the diagonals only
3 x 3 x 3	14-way	2782	Faces + diagonals

Problem No. 7 (PC3-13) was to find values of A and B such that

$$\sqrt{A}^{\sqrt{B}}$$

is as close as possible to an integer.   Mr. Harry Nelson, Livermore, California, furnished the pair 3516658 and 5, for which the power function is 20835119.00000002727275548. Mr. Nelson states that this is the best pair obtainable in the region

$$2 \leq A \leq 5000000, \quad \sqrt{A}^{\sqrt{B}} \leq 10^8$$

James Stein, Woodside, California, furnishes the
following proof for the game of Fourway, which was
introduced in issue No. 3.

Theorem: There is no pattern of numbers in cells
that can cause a game of Fourway to loop indefinitely.

Proof: The array of cells is finite; thus, the
number of patterns is finite. Thus, if a game is looping,
there must exist at least one pattern which occurs
repetitively. Let "P" be such a pattern. Let "C1"
be the cell from which the move is to be made on some
occurrence of "P." At least four moves from C1 must be
made before the number in C1 can reoccur, and P cannot
reoccur until the number in C1 reoccurs.

Let C2 be the cell in direction "1" from C1. Then
a move into C2 must occur before P reoccurs. But
the move into C2 changes the number there; thus, 4 moves
from C2 must also be made before P can reoccur. By
induction, there exists an infinite sequence of cells,
C1, C2, C3,...,C_n,..., where C_{i+1} lies in the "1"
direction from C_i. But this violates the fact that
the array of cells is finite.

Some further solutions to the Four 4's Problem,
given in PC2-1 have been furnished by Rollin Sattler:

$$\frac{\frac{4!}{.4} + \sqrt{4}}{.4} = 155$$

$$\frac{4^{\sqrt{!4}}}{.4} - \sqrt{!4} = 157$$

$$\frac{4^{\sqrt{!4}}}{.4} - \sqrt{4} = 158$$

$$\frac{4^{\sqrt{!4}}}{.4} + !\sqrt{4} = 161$$

$$(4 + \sqrt{!4})\ 4! - \sqrt{!4} = 165$$

$$(4 + \sqrt{!4})\ 4! - \sqrt{4} = 166$$

Desk Calculator Review

Compucorp 320 and 340

Computer Design Corporation of Santa Monica makes six models of portable (battery and/or AC) calculators. Reviewed here are the 320 (scientific) and 340 (statistical) machines; a later review will cover the 322, 324, 342, and 344 machines, which are programmable. The 320 sells for $695; the 340 for $795.

All calculations are carried to 13 digits, but only 10 digits are displayed. Floating decimal operation is automatic, but the display can be set to show from zero to 9 decimal places.

A unique feature is the inclusion of parentheses as operators; a nest of inner parentheses can be made within outer parentheses.

The results in the N-series for 5 (PC5-3) all checked to the limit of the machine.

All entries and results are held in storage until erased. This gives the effect of a constant in all operations, so that repeated operations with one operand constant are particularly easy to perform. Operations can be performed into the 10 storage registers, giving what the makers call Direct Register Arithmetic.

The 340 statistical model takes off the trig functions and the degree/radian functions, and adds the following statistical functions: mean, standard deviation, linear regression, product-moment correlation, t test (dependent and independent), and Z test.

The internal function calculations are quite fast (about one second), with no blinking in the display. The logic of the machines is algebraic (i.e., 7/2 is done by depressing 7, ÷, 2, and = in that order).

These are well designed machines. The 320 manual is a model of clear English, written by someone who knows how to use the machine.

The scientific 320 is a portable, briefcase-sized machine (5.5 x 9 x 2 inches). In addition to the arithmetic operations, it has the trigonometric functions; logs and antilogs to both bases; polar to rectangular coordinates; degree conversion; power function; square root and reciprocal; and 10 storage registers.

7

Log 7	0.845098040014256830712216258592636193483572396323397
Ln 7	1.945910149055313305105352743443179729637084729581686
$\sqrt{7}$	2.645751311064590590501615753639260425710259183082845
$\sqrt[3]{7}$	1.912931182772389101199116839548760282862439050344588
$\sqrt[5]{7}$	1.475773161594552069276916695632244106544093613741020
$\sqrt[7]{7}$	1.320469247756123791809327331500263082736600151497336
$\sqrt[10]{7}$	1.214814044039066939398747361405091290718388035066413
$\sqrt[100]{7}$	1.019649663856845912662824394842508330655450159198699
e^7	1096.633158428458599263720233288121432442219134833361
π^7	3020.293227767920675142064930720418319174324752954O
$\tan^{-1} 7$	1.428899272190732696418470074537198359090802940955909
7^{100}	3234476509624757991344647769100216810857203198904625 4009338953313916914596369280600001
7^{1000}	1253256639965718318107554832382734206164985075080986 1714634950075209705963173811643244883905435152076319 8615919551594076685828989467263022761790838270854579 8300151112466612039846243589298325716157180147040963 0566809750761327366302322689525054138592715842608868 4494082416768617708189592286936039922311125683719215 0466891567383525901372415545101858559645499275754932 4739113254853437849797880608495108587420201183636231 5727420109554782988791530088289711844550500230485638 4131899471321422439473341992593007356224929374194536 5006149030210512792031443040163685567754913633748132 1811349678427076091437345045399337348611261168055929 3554029928231924911903600270361122831809358727752145 1746401317827465710073632156460683852739601156414628 4455436631446960506501608126218143270626661951727017 80200286645023823083185928061371310300829284071141 2 07731280600001

N-SERIES

A Dice Problem PROBLEM 16

     In a game that is currently being marketed, a player
starts with a list of the numbers from 1 to 9.   He throws
two dice repeatedly, and for each throw he crosses off
the total shown by the dice from the numbers remaining
in his list.   When he can no longer continue, play
passes to the next player, and the score for the first
player is the sum of the numbers not crossed off.   Low
score wins.

     In the list below, the same sequence for the dice
is played four different ways, with markedly differing
scores.   It would seem that there is a strategy of play.

```
        1  2  3  4  5  6  7  8  9

     5              X
     7                 X
    10        X     X
     3     X
     6   play ends; the score is 20
    ─────────────────────────────────
     5           X
     7        X  X
    10   X                       X
     3   play ends; the score is 23
    ─────────────────────────────────
     5      X  X
     7   X              X
    10   play ends; the score is 33
    ─────────────────────────────────
     5              X
     7                 X
    10   X                       X
     3      X
     6               X
    12         X           X
     2   X
     7   all numbers are crossed off;
         the score is zero.
```

   (1)   What is the proper strategy in this game?
   (2)   Write a program to play the game; the input to
the program consists of successive dice throws; the output
is the choice of numbers in the 1-9 series to be crossed
off.

Squares on a Lattice PROBLEM 17

On an NxN array of points on a Cartesian grid, how many sets of 4 points can be found that form squares?

For the 2x2 case, there is only one set, as in (1) in Figure M.

For the 3x3 case, there are four sets of the type shown in (2); one set as in (3); and one set as in (4); for a total of 6.

For the 4x4 case there are nine sets as in (5); four sets as in (6); one set as in (7); four sets as in (8); two sets as in (9); for a total of 20.

Similar analysis shows a total of 49 sets for the 5x5 case, and 86 for the 6x6 case.

Write a program to calculate the number of squares that can be found on an N x M lattice.
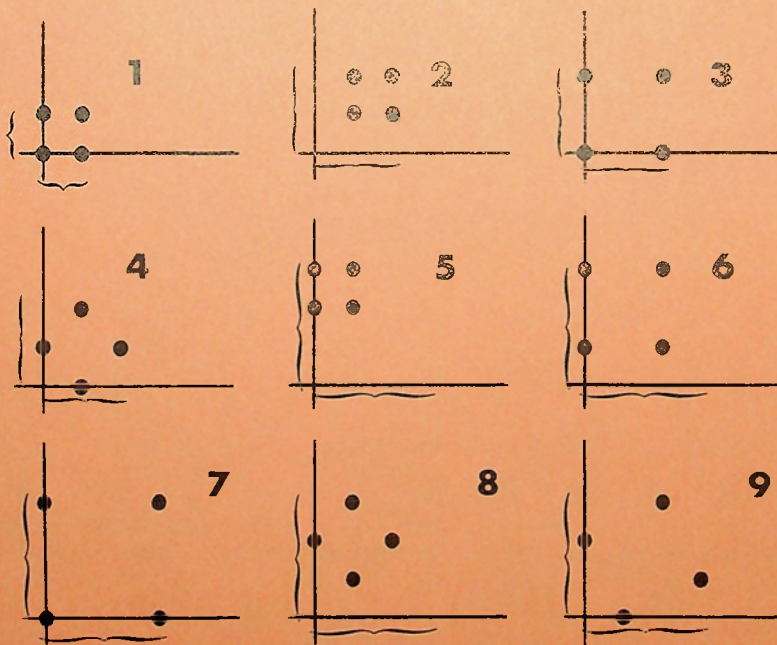
M

Lion Hunting

The article, "A Contribution to the Mathematical Theory of Big Game Hunting," by H. Pétard (pseud.) appeared in the August/September 1938 American Mathematical <u>Monthly</u>.   One of the methods given there was:
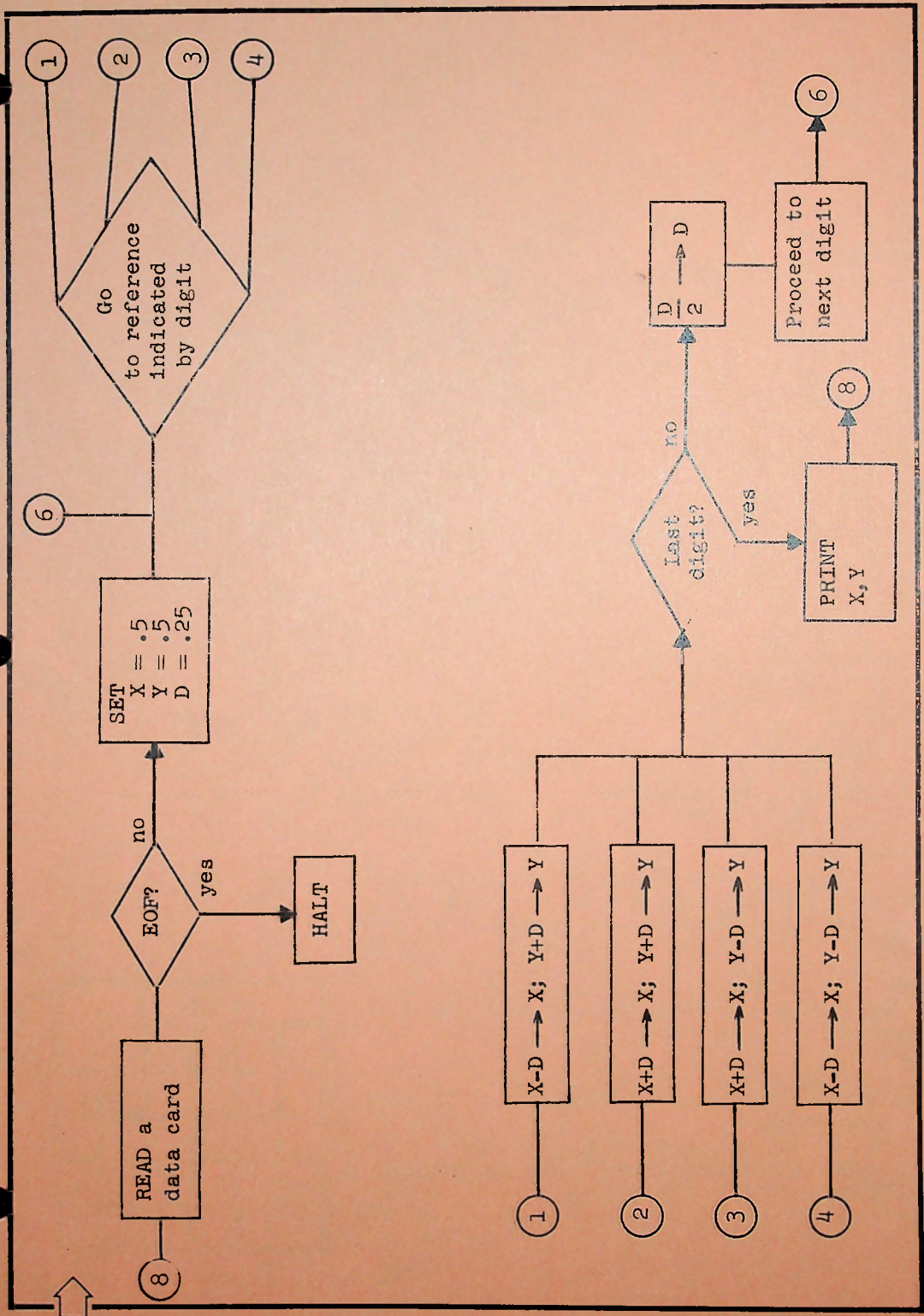
The Bolzano-Weierstrass Method.   Bisect the desert by a line running N-S.   The lion is either in the E portion or in the W portion; let us suppose him to be in the W portion.   Bisect this portion by a line running E-W.   The lion is either in the N portion or in the S portion; let us suppose him to be in the N portion. We continue this process indefinitely, constructing a sufficiently strong fence about the chosen portion at each step.   The diameter of the chosen portions approaches zero, so that the lion is ultimately surrounded by a fence of arbitrarily small perimeter.

If the four quadrants of the desert (the whole desert being taken as a unit square) are numbered 1 for northwest, 2 for northeast, 3 for southeast, and 4 for southwest, then a series of 32 digits will locate the lion to ten digit precision.   See the Figure, where the first steps in the sequence 31242...are shown.
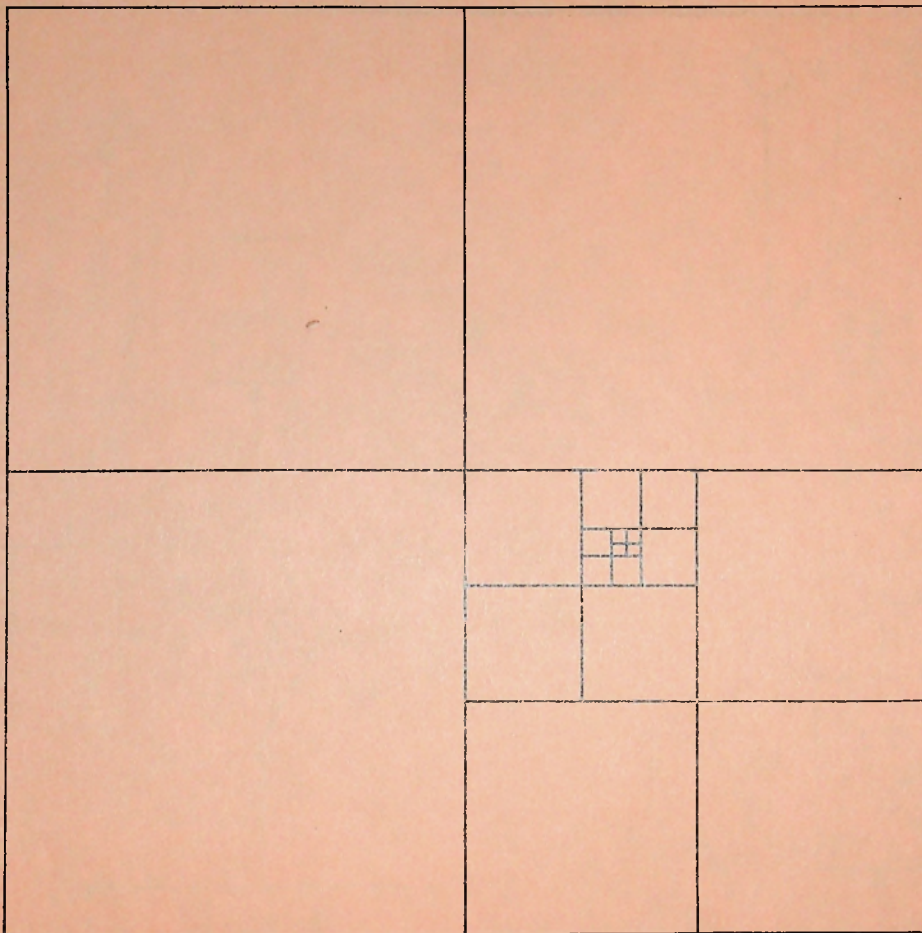
The flowchart shows the logic for accepting 32 digits and locating the lion.   Some test cases are given here:

Moves	X	Y
11111111111111111111111111111111	.0000000000	1.0000000000
22222222222222222222222222222222	1.0000000000	1.0000000000
33333333333333333333333333333333	1.0000000000	.0000000000
44444444444444444444444444444444	.0000000000	.0000000000
12341234123412341234123412341234	.4000000000	.8000000000
14141414141414141414141414141414	.0000000000	.6666666667
14444444444444444444444444444444	.0000000000	.5000000000
23333333333333333333333333333333	1.0000000000	.5000000000
11444444444444444444444444444444	.0000000000	.7500000000
34421123344314321422124313214412	.5764649737	.1173360766
11223344112233441122334411223344	.2352941177	.9411764705
43214321432143214321432143214321	.4000000000	.2000000000

READ a data card

EOF? — yes → HALT
no

SET
X = .5
Y = .5
D = .25

Go to reference indicated by digit

1 2 3 4

6

1 → X−D → X; Y+D → Y
2 → X+D → X; Y+D → Y
3 → X+D → X; Y−D → Y
4 → X−D → X; Y−D → Y

last digit? — yes → PRINT X, Y → 8
no

D
─ → D
2

Proceed to next digit → 6

8

The start of a lion hunt with the digits 31242...

The pattern shown here is a common design for linoleum and tile. A 6 x 6 square is divided up into eleven 1 x 2, two 1 x 1, and two 2 x 2 blocks.

Problem: In how many distinct ways can the 15 pieces be placed on the 6 x 6 grid?

Is this a computer problem? If so, a method of attack is needed, and this is PROBLEM 18.

If not, a rationale is needed for producing the result by other means.